Computing Systems Services Technical Bulletin

Number:	0052
Issued Date:	June 21, 1984
Effective Date:	June 20, 1984
Section/Groups:	
Submitted By:	
Approved By:	

NATURAL System Maintenance 5

A new release of NATURAL was installed on MVS2 on June 19, 1984 and on MVS1 on June 20, 1984. The following enhancements and modifications are included in the release and are now available on both systems.

%* Is Now Recognized In Batch Input

When executing a NATURAL session in batch, the command string "%*" is recognized, where "%" is the current terminal control character as defined by the keyword "CF". If data is read from either CMSYNIN or CMOBJIN which contains this command string in the first two columns, the input data which follows on the next input record will not be printed on CMPRINT. This function can be used to disable the printing of passwords and other sensitive information.

Example:

```
//CMSYNIN DD *
command
%*
value
/*
```

The above will result in only the command and not the value being printed.

Stack (TOP) (DATA) (FORMATED)

The "STACK DATA" statement has been extended to include a new option. The option "FORMATED" permits the transfer of data through the NATURAL stack transparently. Data stacked with this option will be passed to the next executed INPUT statement without modification.

The default option DELIMITED will process the data in DELIMITER mode with the consequence that embedded delimiter characters and INPUT assign characters will be interpreted.

Find ... and Retain as Same Name

It is now possible to use the same name in the RETAIN clause as in the basic search criteria.

```
FIND FILE WITH ‘LIST1' ... RETAIN AS ‘LIST1'
```

Mask Facility

There have been some changes in the data validation part of the NATURAL MASK facility. These changes are as follows:

1. The default date value is day one of the current year.

2. When validating a year value any year is accepted, not just the current year.

3. The data consistency check is now performed at the end of the mask processing to permit interspersed characters such as "/" and ":".

Global Variables

Beginning with SM05, it is possible to catalog a NATURAL program which contains source global variables (i.e., variables ot the type &VAR). At the time the program is cataloged, the current value of a source global variable will be used. It is, therefore, not possible to reconstruct the source program at any time.

When a NATURAL program is cataloged, the value of alphanumeric global variables is set to binary zeros. Thus, when such a global variable is used for the first time in a sequence of programs, its initial value will not be blanks but binary zeros. It has been necessary to do this to ensure that ADABAS retains all occurrences of the field in which the NATURAL program is stored. It is intended that a future SM release will contain the modification necessary to restore the initial value of alphanumeric global variables to blanks.

Read Logical Statement

The usage of a multiple valued descriptor to define the logical sequence in a READ statement which includes the "THRU" or "ENDING AT" clause will now generate a NAT0160 syntax error. The reasoning is that a multiple valued descriptor does not define a reproductive sequence and hense the termination condition is unpredictable.

MOVE INDEXED

NATURAL will no longer allow a MOVE INDEXED when the index has a value of zero. A NAT1311 will be returned.

Binary Descriptions

Usage of binary descriptions in FIND statements which included an implicit upper range value being generated by NATURAL gave incorrect results (e.g., FIND FILE WITH BINDESCR GT VALUE). The upper range value was always a value of X’EF’ with leading binary zeros if required. A true value of X’EF’ to the length of the descriptor value is now generated.

Referencing Data Base Fields

The use of multiple FIND and/or READ statements will generate a hierarchy of processing loops. When such multiple processing loops are active, the fields contained in all referenced files/userviews are available for processing simply by specifying the field name. An explicit occurrence of a data base field can be specified by supplying a reference with the field, for example, FLD (0050). NATURAL resolves the reference of a data base field if no explicit reference is provided. This is achieved by scanning all active userviews for the specified field name until the first occurrence is located. The hierarchical order in which the userviews are scanned is as follows:

1. The currently active userview from the innermost loop.

2. The hierarchy of active userviews is stepped outwards from the innermost (first) to the outermost (last).

3. If a userview is introduced with a non-loop generating statement (e.g., FIND FIRST), then this userview is placed in the hierarchy according to its position in the loop structure.

There is an error that cannot be resolved until the next system modification level which causes the above algorithm to fail in certain circumstances. Automatic referencing will give the wrong result in the following case:

1. The data base field name "FLD" does not exist in the currently active userview.

2. The field does exist in a userview further out in the hierarchy of active userviews. (These userviews could all be the same or different userviews).

3. An occurrence of the field has been referenced from one of the userviews in the active hierarchy.

In the above situation the reference is resolved from the first occurrence found and not from that userview in the active hierarchy which is closest to the innermost loop.

Example:

```
0010    FIND FILE1
0020            WRITE FLD
0030                    FIND FILE1
0040                            FIND FILE 2
```

```
0050                    WRITE FLD
```

The field "FLD" exists in userview "FILE1" but not in "FILE2". The occurrence of FLD in statement (0050) should be resolved by the FIND at statement (0030) as specified in the algorithm above. However the reference will be resolved by the FIND at statement (0010) because of the occurrence of FLD in statement (0020).

New Functions fo the Full Screen Editor

The following enhancements have been made to the NATURAL full screen editor:

X-Line on Different Screen

For shift operations, the X-line is not required to be on the current screen.

Renumbering Lines

The command "REN ON" will cause source lines to be renumbered automatically if any of the following conditions are true:

C A new line is to be inserted and there is no line number available between the current line and the next line.

C A STOW command is issued for a program.

C A CAT command is issued for a program.

The "REN ON" setting may be set to off by specifying "REN OFF".

Inserting Programs

The line command ".I(program)" causes the named program to be inserted into the source area after the current line. The program in the source area is then renumbered. The program to be inserted must be in the current library or in the library "SYSTEM".

PF And PA Key Profile Setting

The command "PROFILE" causes a map to be displayed with which a profile for PF/PA key setting may be created and/or modified. The creation/modification of the PF/PA key settings will be restricted if NATURAL security is in use.

The command "PROFILE profile-name" caused the specified PF/PA key profile to be activated.

Split Screen Enhancement

Capability to view a DDM or a second program on the lower portion of the screen while editing a program is now available. The following commands control this function and can be issued from the EDITOR command line:

Command Fuction

‘F FILE-NAME’ List DDM on bottom 10 lines of screen

‘FS FILE-NAME’ List short form of DDM

‘FB’ ‘FT’ List first (or last) 9 lines of DDM

‘FSC NAME’ Scans for ADABAS short name or long name in DDM and marks the line

‘F+nn’ ‘F-nn’ Rolls the screen nnn lines forward or backward

‘F+P’ ‘FP’ ‘F-P’ Rolls the screen one page forward or backward

‘F+H’ ‘FH’ ‘F-H’ Rolls the screen ½ page forward or backward

‘FE’ ‘F END’ ‘FE’ Reverts to full screen mode editing

‘P NAME LIB ‘ Lists a program on bottom 10 lines of screen

‘PSC VALUE’ Invokes scan for "VALUE". Functions the same as the SCAN command of the editor. As long as "P-SCAN" is indicated on the screen, use of the "ENTER" key will cause a scan to the next marked line of the program being listed. If an editor command is entered, it will be processed.

‘PNNNN’ Positions the listed program at source line nnnn.

‘P+P’ ‘PP’ ‘P-P’ Positions the listed program forward or backward one page.

‘P+H’ ‘PH’ ‘P-H’ Positions the listed program forward or backward ½ page

‘P+NNN’ ‘P-NNN’ Positions the listed program forward or backward nnn lines.

‘PB’ Lists the last 9 lines of the program.

‘PT’ Lists the first 9 lines of the program.

‘PE’ and ‘P END’ Reverts the screen back to full screen editing mode.

Note: A NAT0888 may be received when a Split Screen function is invoked and many Globals are present. This occurs because the Split Screen Editor uses many Global Variables which are not reset within the Full Screen Editor.

Security By Value Processing In NATURAL

The following describes security by value processing as it relates to NATURAL statements.

If the number of records selected as the result of a NATURAL FIND statement is not equal to 0 (ADABAS ISN Quantity not equal 0), the system variable *NUMBER is set to "high-values" (999.999.999) to indicate that one or more records met the selection criteria. The user may not be permitted to read each or any of the records selected because of security by value restriction.

Beginning with NATURAL SM05, the system variable *ISN will be set to 0 at the end of the FIND loop (or after a FIND FIRST statement) if the user was not permitted to read any of the selected records. The user may, therefore, interrogate *ISN to determine if any records were actually read.

The system variable *COUNTER contains the number of non-protected records processed as a result of a FIND loop of FIND with WHERE clause.

FIND NUMBER with a WHERE clause will cause "high values" to be returned in the system variable *NUMBER and the number of non-protected records in the system variable *COUNTER.

FIND NUMBER without a WHERE clause should not be used with security by value protected files since the actual number of non-protected records for the user is only determinable by reading each record.

For further information on any of the above changes, please contact Sandra Neal at 533-4040 or Larry Barker at 533-5302.